**NAT Feature Test Cases list (FT\_IT)**

**Feature Overview:**

NAT allows an organization with non-globally routable addresses to connect to the Internet by translating those addresses into globally routable address space.

NAT is configured at the exit device in a domain. When a packet exits the domain, NAT translates the locally significant source address into a globally unique address. When a packet enters the domain, NAT translates the globally unique destination address into a local address.

**NAT Use Case (From Microsoft XBOX NAT Requirements) :**

In an effort to make Xbox gaming consoles available as a cloud service, the Xbox team is considering hosting thousands of custom Xbox servers within Azure datacenters. These consoles will provide public endpoints for user to access. However, due to the large number of such consoles, Azure cannot provide a public IP for every Xbox console. Therefore, a NAT solution is required to save the public IPs while providing access to those consoles from public.

**Feature Testing:**

**Functionality:**

1. Verify static NAT establishes a one-to-one mapping between the inside local address and an inside global address.
2. Verify dynamic NAT establishes a mapping between an inside local address and an inside global address dynamically selected from a pool of global addresses.
3. Verify dynamic NAPT (PAT) functionality with ACL binding to NAT Pool, that allow only a specific set of hosts to get translated with IP protocol type as TCP.
4. Verify dynamic NAPT (PAT) functionality with ACL binding to NAT Pool, that allow only a specific set of hosts to get translated with IP protocol type as UDP.
5. Verify dynamic NAPT (PAT) functionality without ACL binding to NAT Pool, that allow all set of hosts to get translated with IP protocol type as TCP.
6. Verify dynamic NAPT (PAT) functionality without ACL binding to NAT Pool, that allow all set of hosts to get translated with IP protocol type as UDP.
7. Verify static NAPT (PAT) functionality for TCP and UDP traffic with separate static NAPT entries configured for each protocol type.
8. Verify that dynamic NAPT entry timeout for TCP traffic with default & non-default timeout value, when no active traffic forwards for that NAT entry.
9. Verify that dynamic NAPT entry timeout for UDP traffic with default & non-default timeout value, when no active traffic forwards for that NAT entry.
10. Verify that dynamic NAPT entry doesn’t time out and there is no traffic disruption when traffic is actively forwarding for a duration that is longer than the configured time out.
11. Verify that static NAT entries are not timed out when there is no active traffic forwarding, and the get deleted only when explicitly removed.
12. Verify that no traffic disruption occurs when a dynamic NAPT entry is converted to a static entry if static NAPT entry is created for it.
13. (i) Verify the static NAT & NAPT functionality works fine after config reload.

(ii) Verify the dynamic NAT & NAPT functionality works fine after config reload.

1. (i) Verify the static NAT & NAPT functionality works fine after removing and re-apply the configuration.

(ii) Verify the dynamic NAT & NAPT functionality works fine after removing and re-apply the configuration.

1. Verify dynamic NAT functionality with ACL binding to NAT Pool with multiple NAT Pools.
2. Verify dynamic NAPT (PAT) functionality with ACL binding to NAT Pool with multiple NAT Pools.
3. Validate the ACL modifications are gracefully handled when modification are done after ACL-NAT pool binding, and while corresponding active NAT traffic is forwarding.
4. Validate static & dynamic NAT functionality after restarting (stop & start) the NAT Docker service.
5. Validate static & dynamic NAPT functionality after restarting (stop & start) the NAT Docker service.
6. Verify the dynamic NAPT functionality after removing and reapplying the ACL-NAT Pool binding.
7. Verify the dynamic NAPT functionality by first configuring the NAT Pool, have some NAT entries learning, then configure ACL and bind it to NATP Pool and check the new NAT learning.
8. Verify the static NAT & NAPT functionality by first configuring the static NAT entry and then configure the Routing interfaces.
9. Verify the dynamic NAT & NAPT functionality by first configuring the NAT Pool and then configure the Routing interfaces.
10. Verify dynamic NAPT functionality after changing the inside host ipv4 addresses from one physical port to another port.
11. Verify dynamic NAPT functionality after changing the inside routing interface from a physical port to a vlan routing interface (on the same physical port).
12. Verify dynamic NAPT functionality after changing the inside routing interface from a physical port to a Port-Channel routing interface.
13. Verify dynamic NAPT functionality after changing the inside routing interface from a Port-Channel routing interface to a vlan routing interface (on the same Port-Channel interface).
14. Verify the dynamic NAPT functionality after removing and reapplying the global ipv4 address on the outside interface.
15. Verify the dynamic NAPT functionality after removing the global ipv4 address (related to NAT Pool -1) on the outside interface and configure a new IPv4 global ipv4 address (related to NAT Pool-2).
16. Verify the miss configuration of global ipv4 address on the outside interface is gracefully handled and dynamic NAPT works fine after correcting the ipv4 address as per NAT Pool.
17. Verify the miss configuration of zone is gracefully handled and dynamic NAPT works fine after correcting the zone configurations on inside and outside interfaces.

Note -

1. Both Vlan based and Port based routing interfaces are covered across the test cases
2. Zone/realm configuration is implicit in all test cases.

**Platform Specific:**

1. Verify Static NAT functionality on all supported platform switching silicons.
2. Verify Dynamic NAT functionality on all supported platform switching silicons.

**Warm Boot:**

1. Verify the Static NAT functionality works fine during and after Warm boot with active traffic forwarding (expecting 0% loss). Verify that all NAT entries are retained post warm boot.
2. Verify the Dynamic NAT functionality works fine during and after Warm boot with active traffic forwarding (expecting 0% loss). Verify that all NAT entries are retained post warm boot.
3. Verify the Static NAPT functionality works fine during and after Warm boot with active traffic forwarding (expecting 0% loss). Verify that all NAT entries are retained post warm boot.
4. Verify the Dynamic NAPT functionality works fine during and after Warm boot with active traffic forwarding (expecting 0% loss). Verify that all NAT entries are retained post warm boot.
5. Verify that new NAT translation entries are getting installed into Hardware and traffic is forwarded successfully after warm-boot.

**Cold Boot:**

1. Verify the static and dynamic NAT functionality works fine after cold boot. Verify that all NAT entries are retained post cold boot.
2. Verify the static and dynamic NAPT functionality works fine after cold boot.

Note - We will cover this cold boot scenario in some of the above functional test cases.

**Manageability:**

1. Verify the CLI commands of the feature are working fine.
2. Verify other supported manageability of the device ( JSON, REST API etc.)

**Serviceability:**

1. Verify that logging for NAT can be enabled at al supported levels and log messages are getting generated in corresponding scenarios.
2. Verify all debug commands related to NAT are working as expected.
3. Verify all different counters in NAT stats table (per global level and zone level) are getting updated properly in all valid scenarios.
4. Verify the NAT stats table entries can be cleared and updated per global level and zone level.
5. Verify all different fields in NAT translation table (per global level and zone level) are getting updated properly in all valid scenarios.
6. Verify the NAT translation table entries are cleared successfully and updated again if active traffic is forwarding.
7. Verify scaling beyond table limits (per max pool) and ensure the 'Table full' condition is handled gracefully and reported.
8. Verify the error handling case in case of invalid time-out value, invalid IP address pool, invalid port numbers pool.

**Functional Interaction:**

1. Validate NAT functionality works on Port-channel interface configured as Port based routing interface.
2. Validate NAT functionality works on Port-channel interface configured as Vlan based routing interface.
3. Validate the NAT functionality by first configuring ACL on Port-channel, ACL-NAT Pool binding and then add port members to that Port-channel.
4. Verify the dynamic NAPT functionality on Port-channel, after removing the present member ports and add new member ports.
5. Verify ACL permit and deny rule working on an interface configured with inside ( port connected to host inside the network) network for NAT ( for NAT related traffic and traffic not related to NAT )
6. Verify ACL permit and deny rule working on an interface configured with outside ( port connected to external network) network for NAT ( for NAT related traffic and traffic not related to NAT ).
7. Validate traceroute functionality (with NAT) from inside local address (host) to outside global address and vice-versa.
8. Validate Ping functionality (with NAT) from inside local address (host) to outside global address and vice-versa.

**Stress:**

1. Verify the shutdown and no shutdown of the gateway interface used at the border for network address translation -- repeatedly and verify that the NAT functionality works fine after those iterations. Also, verify that the DUT is stable. Also, verify that there are no memory leaks.
2. Have the dynamic NAT entry time-out to lowest possible value and then create a scenario there is frequent learn / unlearn ; config / unconfig of NAT entries over some duration of the time. In this case, verify that the DUT is stable, no memory leaks, no unwanted logs.
3. With dynamic NAT enabled, send line rate traffic corresponding to dynamic NAT learning. Verify that NAT learning pkts going to CPU are put into the right CPU queue. Verify that CPU rate limiting comes into picture w.r.to NAT learning and there should not be any memory leaks and DUT is stable.
4. Verify that there are no memory leaks when NAT feature is configured and active for 12 hours.
5. Verify that there are no memory leaks when NAT feature is configured and unconfigured repeatedly for 10 iterations with traffic forwarding.
6. Make the CPU busy with continuous ping / SNMP root node walk / line rate mcast / bcast storm. Now, send NAT related traffic and verify that NAT learning is successful.
7. Verify the DUT stability and NAT learning rate when user tried to learn the max supported NAT entries in a single burst at line rate. - This test needs to be executed on all supported platforms.

**Negative Test:**

1. Verify that same interface can not be used for inside network and outside network in NAT.
2. In case of static NAT, if there is an overlap of address range, verify that DUT handles that gracefully.
3. In case of dynamic NAT, if there is an overlap of address range, verify that DUT handles that gracefully.
4. Verify that the pool deletion command is thrown error if the pool is already in use by at least one L3 interface.

**Scaling:**

1. Validate max number of concurrent NAT session (in dynamic NAT) supported on the DUT.
2. Validate max number of NAT pools supported on DUT.
3. Verify the max number of static NAT session supported on the DUT.
4. Verify that timed-out entries are creating space for new NAT entries and again limited to maximum entries.

**Performance:**

1. Verify the HW installation time for max no. of NAT translated flows.
2. Verify the traffic throughput for max no. of NAT translated flows.

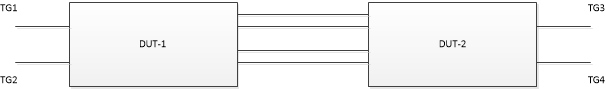
**Traceability Matrix (Feature Requirement vs Test case ID mapping):**

Note - New requirements may get added after QA review comments on design spec are addressed.

|  |  |  |
| --- | --- | --- |
| **Req.no.** | **Req. description** | **Test Case ID** |
| 1.1.1 | Provide ability to create/delete Static Basic NAT entries (one-to-one IP address translation) mapping from public IP address to an internal host's IP address. | 1 |
| 1.1.2 | Provide ability to create/delete static NAPT (PAT) entries that map an L4 port on the Router's public IP address to an internal host's IP address + L4 port. | 7 |
| 1.1.3 | Provide ability to do dynamic NAT from internal host IP addresses to the available range of public IP addresses. | 2 |
| 1.1.4 | PAT entries are configurable per IP protocol type. Allowed IP protocols are TCP and UDP. | 3-9  (All test cases covers either TCP/UDP) |
| 1.1.5 | Configure NAT pool that specifies the range of IP addresses and range of L4 ports to do dynamic network address translation to. | 2-4 |
| 1.1.6 | More than 1 NAT pool can be created limited to a maximum number of 16 pools. |  |
| 1.1.7 | Access lists are used to define the set of hosts that are subjected to dynamic NAT/NAPT, by binding ACL and NAT pool together. | 3,4  (Most of the tests covers the ACL-Pool binding) |
| 1.1.8 | ACL and NAT pool binding is applicable on the L3 ports defined by the ACL TABLE entry. | 3,4  (Most of the tests covers the ACL-Pool binding) |
| 1.1.9 | NAT pool binding with no associated ACL, allows all hosts to be subjected to dynamic NAT/NAPT. | 5,6 |
| 1.1.10 | The L3 ports on which the NAT ACL is applied are in a different NAT zone compared to the port on which the NAT Pool IP address is based on. | ~~51~~ 66 |
| 1.1.11 | For the NAT/NAPT entries created statically or dynamically, bi-directional NAT translations can be performed. | All tests covers the bi cirectional traffic. |
| 1.1.12 | Provide configurable age timeout interval for the inactive UDP NAT entries (in seconds). Default is 300 secs. Range is from 120 sec to 500 secs. | 9 |
| 1.1.13 | Provide configurable age timeout interval for the inactive TCP NAT entries (in seconds). Default is 86400 secs. Range is 300 sec to 432000 secs. | 8 |
| 1.1.14 | Provide configuration of outside zones/realms on L3 interfaces. | Will be covered in CLI testing and al test cases implicitly uses this configuration |
| 1.1.15 | Provide support for translation statistics per zone and per NAT flow. | ~~33-36~~  45-48 |
| 1.1.16 | Dynamic and static NAT/NAPT entries should persist across warm reboot with no traffic disruption to the active flows. | ~~22-26~~  34-38 |
| 1.1.17 | Support NAT configuration from JSON file. | ~~30~~  42 |
| 1.1.18 | Support NAT configuration via incremental CLI. | ~~29~~  41 |
| 1.1.19 | Ability to clear the NAT translation table entries. | ~~35-36~~  47-48 |
| 1.1.20 | Ability to clear the NAT translation statistics. | ~~33-34~~  45-46 |
| 1.1.21 | Dynamic NAPT entry is timed out if it is inactive in the hardware for more than the configurable age timeout period. | 8-10 |
| 1.1.22 | Static NAT/NAPT entries are not timed out. They have to be unconfigured explicitly. | 11 |
| 1.1.23 | If Static NAPT entry is same as the dynamic NAPT entry, entry is retained as Static NAPT entry. | 12 |
| 1.1.24 | Ability to enable the logging at different severity levels of the NAT module. | ~~31~~  43 |
| 1.1.25 | Ability to stop and start the NAT docker service. | 18, 19 |
| 1.1.26 | The SNAT or DNAT miss packets are rate limited to CPU @ 600pps. | ~~47~~  61 |
| 1.1.27 | The NAT miss packets are processed from a higher priority CPU COS Queue than the Broadcast/Unknown Multicast packets. | ~~47~~  61 |
| 1.1.28 | The hardware NAT table full condition is handled by the OrchAgent gracefully. | ~~37~~ 49, ~~55-57~~  70-73 |
| 1.1.29 | ICMP packets are translated via NAT or NAPT rules in the Linux kernel. | ~~43, 44~~  57, 58 |
| 1.1.30 | Should be able to ping from internal host to an outside host via NAPT. | ~~44~~ 58 |
| 1.1.31 | Should be able to traceroute from internal host to an outside host via NAPT. | ~~43~~ 57 |

**Test Topology**

DUT-1 is the device running NAT feature



Note - We use two DUT topology to cover Port channel interface related tests, other tests can be validated on single DUT.

**Test Gear Requirements:**

4 Traffic Generator Ports